What is claimed is:

1. A multiprocessor computer system comprising:

a first processor segment having a first microprocessor with a first internal cache, the contents of the first internal cache being unknown externally to the first processor segment;

a second processor segment having a second microprocessor with a second internal cache, the contents of the second internal cache being unknown externally to the second processor segment;

main memory;

a first interprocessor bus for coupling the first and second processor segments and the main memory;

a second interprocessor bus for coupling the first and second processor segments and the main memory;

a crossbar switch for switching between the first interprocessor processor bus and the second interprocessor processor bus;

the first processor segment further comprising:

an external cache memory;

a memory controller for causing the storage of data within the external cache memory, the data stored in the external cache memory corresponding to data in the first internal cache;

a memory identifier for identifying whether the data in the external cache memory is up to date relative to the memory in the first internal cache; and

a data director for directing data requested by the second processor segment from the more up to date of the external cache memory and the first internal cache.

2. The multiprocessor computer system as defined by claim 1 wherein the first microprocessor includes an associated invalidate flag that when set, invalidates data in the first internal cache.

3. The multiprocessor computer system as defined by claim 2 wherein the memory controller includes a determiner that determines if data within the first internal cache has been modified by the first microprocessor, the data director directing modified data to the second processor segment.

4. A method for maintaining coherent data in a multiprocessor system having a plurality of processors coupled to main memory, where each processor has an internal cache which is externally unreadable outside the processor, the method comprising:

requesting data associated with a memory location in main memory;

determining if an external cache coupled to an application specific integrated circuit associated with a second processor contains a reference to the requested data;

performing a snoop cycle on the second processor if the external cache has a reference to the requested data; and

determining if the requested data has been modified in the second processor.

5. A method for maintaining coherent data according to claim 4 further comprising:

sending to the requesting processor from the second processor the requested data if the second processor has modified the requested data.

6. A method for maintaining coherent data according to claim 4, further comprising the step of:

recalling from main memory the data associated with a memory location in main memory if theexternal cache fails to have a reference to the requested data.

7. A method for maintaining coherent data in a multiprocessor system having a plurality of processors forming a segment, the processors connected together by a processor bus and the segment connected together by a system bus which is coupled to main memory, where each processor has an internal cache which is externally unreadable outside the processor, the method comprising:

placing a request from a processor in a first segment for data associated with a memory location in main memory on the system bus;

determining if an external cache coupled to any application specific integrated circuit associated with a second segment contains a reference to the requested data; and

performing a snoop cycle on the processor bus of the second segment if the second segment has a reference to the requested data.

8. A method according to claim 7, further comprising:

determining if the requested data has been modified.

9. A method according to claim 8, further comprising:

sending the requested data to the requesting processor from a processor in the second segment if the snoop cycle returns a signal indicating the processor of the second segment has modified the requested data.

10. A method according to claim 7, further comprising the step of:

recalling from main memory the data associated with a memory location in main memory if all of the application specific integrated circuits associated with a segment fail to have a reference to the requested data.

11. A multiprocessor computer system comprising:

a first processor segment having a first microprocessor with a first internal cache, the contents of the first internal cache being unknown externally to the first processor segment;

a second processor segment having a second microprocessor with a second internal cache, the contents of the second internal cache being unknown externally to the second processor segment;

main memory;

a system bus for coupling the first, processor segment, the second processor segment, and main memory;

the first processor segment further comprising:

an external cache memory;

a memory controller for storing address information within the external cache memory, the address information associated with data signals entering and exiting the first processor segment.

12. The system according to claim 11, wherein the first processor segment further includes a processor bus and the second processor segment further includes a processor bus.

13. The system according to claim 11, further comprising:

a second external cache memory; and

a second memory controller for storing address information within the second external cache memory, the address information associated with data signals entering and exiting the second processor segment.

14. The system according to claim 13, wherein the memory controller connects the system bus to the bus of the first processor segment and the second memory controller connects the system bus to the bus of the second processor segment.

15. The multiprocessor computer system as defined by claim 11, wherein the first microprocessor includes an associated invalidate flag that when set, invalidates data in the first internal cache.

16. The multiprocessor computer system as defined by claim 15, wherein the memory controller includes a determiner that determines if data within the first internal cache has been modified by the first processor.